AFC
IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Hari K. Ravichandran		
Assignee:	Sun Microsystems, Inc.		
Title:	Apparatus and Method for Processor Performance Monitoring		
Serial No.:	10/056,244	Filing Date:	January 22, 2002
Examiner:	Aaron D. Matthew	Group Art Unit:	2114
Docket No.:	P2678	Customer No.:	33438

Austin, Texas
October 26, 2005

Mail Stop Appeal Brief - Patents
Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

Dear Sir:

Applicant submits this Appeal Brief pursuant to the Notice of Appeal filed in this case on August 1, 2005. A check is enclosed which includes the $500.00 fee for this Appeal Brief. The Board is also authorized to deduct any other amounts required for this appeal brief and to credit any amounts overpaid to Deposit Account. No. 502264.

I. REAL PARTY IN INTEREST - 37 CFR § 41.37(c)(1)(i)

The real party in interest is the assignee, Sun Microsystems, Inc., as named in the caption above and as evidenced by the assignment set forth at Reel 8846, Frame 0757.

II. RELATED APPEALS AND INTERFERENCES - 37 CFR § 41.37(c)(1)(ii)

Based on information and belief, there are no appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals and Interferences in the pending appeal.

III. STATUS OF CLAIMS - 37 CFR § 41.37(c)(1)(iii)

Claims 9-19 are pending. The rejection of claims 9- 19 is appealed. The Claims Appendix contains the full set of pending claims.

IV. STATUS OF AMENDMENTS - 37 CFR § 41.37(c)(1)(iv)

No amendments after final have been requested or entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER - 37 CFR § 41.37(c)(1)(v)

The present invention, as set forth by independent claim 9, relates to a method for monitoring an execution of a program. The method includes the steps of: (1) obtaining a first indication including a first address (see e.g., page 7, lines 19 – 24 and page 8, line 11); (2) searching a first memory device for an entry associated with the first address (see e.g., page 7, lines 24 – 27 and page 8, lines 11 - 13); (3) when the entry in the first memory device does not exist (see e.g., page 7, lines 27 – 30 and page 8, lines 14 - 15), generating at least one probe signal indicating a miss entry in the first memory device (see e.g., page 8, line15 – 16); (4) generating a temporal identifier signal that is associated with the probe signals (see e.g., page 8, line 16 – 18); and (5) storing the temporal identifier signal and the probe signals in memory (see e.g., page 8, lines 18 – 20).

The present invention, as set forth by independent claim 13, relates to a method for monitoring an execution behavior of a program, which includes generating probe signals representative of memory access misses occurring in a processor (see e.g., page 8, lines 15 – 16), receiving the probe signals and associating a temporal identifier signal with the probe signals(see e.g., page 8, lines 20 – 22), storing the temporal identifier signal and the probe signals, and generating a second high-speed memory miss signal, a second high-speed memory miss count signal and a time stamp signal, the second high-speed memory miss signal indicating a miss in a second high-speed memory, the second high-speed memory miss count signal representing a number of misses in the second high-speed memory, and the time stamp signal indicating when the second high-speed memory miss signal is active (see e.g., page 8, line 24 – page 9, line 19, see also Figure 2 and page 4, line 25 – page 3, line 9).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL - 37 CFR § 41.37(c)(1)(vi)

Claims 9 stands rejected under Bunnell, U.S. Patent No. 5,564,015 (Bunnell) in view of Roeber, et al., U.S. Patent No. 5,682,328 (Roeber). Claim 13 stands rejected under Bunnell, in view of Roeber, in further view of Levine et al., U.S. Patent No. 6,067,644 (Levine). Claim 11 stands rejected under Bunnell, in view of Roeber, in further view of Levine et al., U.S. Patent No. 6,067,644 (Levine).

VII. ARGUMENT - 37 CFR § 41.37(c)(1)(vii)

Independent claim 9 is allowable over Bunnell and Roeber.

Bunnell discloses a central processing unit ("CPU") activity monitor and method providing CPU activity information. The CPU activity monitor includes a timer and an activity event counter for receiving a plurality of mode signals from the CPU, a cache miss signal from a cache memory system, and a clock signal from a clock. An activity-to-inactivity value defines when the CPU transitions from an active state to an inactive state. An activity threshold defines when the CPU transitions from an inactive state to an active state.

More specifically, the Examiner has stated that Bunnell discloses generating a first high speed memory miss signal, a first high speed memory miss count signal and a time stamp signal. The Examiner cites the following portions of Bunnell:

> Thus, the cache memory system 30 generates the cache miss signal 36 when the CPU 28 accesses data that is not stored in the cache memory system 30. (Bunnell, Col. 6, lines 58 – 61.)
>
> The present invention determines CPU activity by counting the number of activity events that occur within a certain time interval. The preferred embodiment defines a CPU activity event as a CPU data write cycle that generates a cache miss signal. (Bunnell, Col. 4. lines 49 – 53.)
>
> The output of the event register 112 and the output of the event counter 114 connect to the event comparator 104. (Bunnell, Col. 9, lines 5 – 6.)
>
> In addition, the CPU activity monitor receives a clock signal. (Bunnell, Col. 4, lines 44-45.)

Roeber, et al. discloses event logging using a single board computer control card configurable onto a backplane containing target processors being monitored. A high resolution

clock on the control card time stamps events. Memory on the control card provides a central buffer to store event data and stores a control program effecting functionality of the card. A network interface facilitates communication with host computers for post processing of event data and to control, communicate with, and access the control card. A control program effects event data collection and organization/storage of events in control card memory. The control program coordinates retrieval of events from an event interface area of memory on slave target processors prior to processing by the control card. The control program coordinates offloading of event data from the control card to host computers for post processing by known software visualization tools. Target software is instrumented with calls to a macro which in turn calls the logging function to effect recording of events. The event records are temporarily stored in an event interface portion of memory on the control card or in a buffer on the target processors.

More specifically, the Examiner has stated that Roeber discloses a method for monitoring and analyzing system activity by recording event data along with time information associated with the event data. The Examiner cites the following portion of Roeber:

> A high resolution clock on the control card is used to time stamp events. A portion of memory on the control card is used as a central buffer to store event data, while another portion of memory on the control card is used to store a control program that effects functionality of the control card. (Roeber, Col. 3, lines 30 – 33.)
>
> The record created will include the time that the event occurred based on the system clock, and may also include some data relevant to the event. (Roeber, Col. 1, lines 33-35.)

The Examiner thus maintains that:

> One of ordinary skill in the art would have been motivated to include a second cache in the method disclosed in Bunnell, in view of Roeber et al, in order to improve system performance. Moreover, one of ordinary skill in the art would have considered it obvious to perform the same steps in analyzing cache miss activity associated with the second memory device as were performed in analyzing cache miss activity associated with the first. (Final Office Action, page 9.)

However, Bunnell and Roeber, taken alone or in combination, do not teach or suggest a method for monitoring an execution of a program which includes *when the entry in the first memory device does not exist, generating at least one probe signal indicating a miss entry in the first memory device*; generating a temporal identifier signal that is ***associated with the probe***

signals; and ***storing the temporal identifier signal and the probe signals*** *in memory*, all as required by claim 9. Accordingly, claim 9 is allowable over Bunnell and Roeber. Claims 10 – 12 depend from claim 9 and are allowable for at least this reason.

Independent claim 13 is allowable over Bunnell, Roeber and Levine.

Bunnell and Roeber are discussed above.

Levine discloses a processor operable for processing an instruction through a plurality of internal stages will produce a result of the processing of the process at each stage or a reason code why the stage was unable to process the instruction. The result or the reason code will then be passed to a subsequent stage, which will attempt to process the instruction. The second stage will forward the reason code when it cannot produce its own result and it is idle. The second stage will create its own reason code when it is not idle but cannot produce a result, and will forward this reason code.

The Examiner has stated that Levine teaches a method of monitoring the execution of instructions in a program including the steps of checking a second cache in the event that an entry in the first cache dies not exist. The Examiner cites to the following portion of Levine:

> Most computer memory systems take advantage of this fact by incorporating small staging areas. These staging areas store frequently used data. Such areas are usually smaller and more rapidly accessed than system memory. This allows systems to complete work faster and thus have higher performance. Similarly, some items held in the staging areas are more frequently used than others. This leads to the use of an additional secondary staging area. If the required datum is not in the first staging area, the second staging area is next checked. If the item is not in the second staging area, system memory is checked. Because of the high probability of finding a required datum in some staging area, the average time to retrieve the datum is typically lower in hierarchically configured memory systems. Consequently, current memory systems are structured as hierarchies of staging areas where the staging areas become larger and slower in order of access.
>
> It is clearly desirable to keep the most frequently reused data in the staging areas (hereinafter referred to as "caches") closets to point of usage. (Levine, Col. 1, line 51 – Col. 2, line 3.)

However, it does not follow that it would have been obvious to analyze cache miss activity for multilevel or multiple caches as claimed merely because Levine discloses two levels of caches.

More specifically, Bunnell, Roeber and Levine, taken alone or in combination, do not teach or suggest a method for monitoring an execution behavior of a program, which includes *generating a second high-speed memory miss signal,* ***a second high-speed memory miss count signal*** *and a time stamp signal, the second high-speed memory miss signal indicating a miss in a second high-speed memory,* ***the second high-speed memory miss count signal representing a number of misses in the second high-speed memory,*** *and* ***the time stamp signal indicating when the second high-speed memory miss signal is active,*** all as required by claim 13. Accordingly, claim 13 is allowable over Bunnell, Roeber and Levine. Claims 14 - 19 depend from claim 13 and are allowable for at least this reason.

Dependent claim 11 is allowable over Bunnell, Roeber and Levine.

Bunnell, Roeber and Levine are discussed above.

As discussed above, the Examiner maintains that Levine teaches a method of monitoring the execution of instructions in a program including the steps of checking a second cache in the event that an entry in the first cache dies not exist.

However, it does not follow that it would have been obvious to analyze cache miss activity for multilevel or multiple caches as claimed merely because Levine discloses two levels of caches.

More specifically, Bunnell, Roeber and Levine, taken alone or in combination, do not teach or suggest before the storing of the temporal identifier searching *a second memory device for an entry associated with the first address,* when the entry in the second memory device does not exist, ***generating at least one probe signal indicating a miss entry in the second memory device,*** and *generating a temporal identifier signal that is associated with the probe signal*

VIII. CLAIMS APPENDIX - 37 CFR § 41.37(c)(1)(viii)

A copy of the pending claims involved in the appeal is attached.

IX. EVIDENCE APPENDIX - 37 CFR § 41.37(c)(1)(ix)

None

X. RELATED PROCEEDINGS APPENDIX - 37 CFR § 41.37(c)(1)(x)

None

XI. CONCLUSION

For the reasons set forth above, Applicant respectfully submits that the rejection of pending Claims 9 - 19 is unfounded, and requests that the rejection of claims 9 - 19 be reversed.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Appeal Brief – Patents, Board of Patent Appeals and Interferences, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450, on October 26, 2005.

Attorney for Applicant — Date of Signature: 10/26/05

Respectfully submitted,

Stephen A. Terrile
Attorney for Applicant
Reg. No. 32,946

CLAIMS APPENDIX

9. A method for monitoring an execution of a program, the method comprising the steps of:

(1) obtaining a first indication including a first address;

(2) searching a first memory device for an entry associated with the first address;

(3) when the entry in the first memory device does not exist, generating at least one probe signal indicating a miss entry in the first memory device;

(4) generating a temporal identifier signal that is associated with the probe signals; and

(5) storing the temporal identifier signal and the probe signals in memory.

10. The method of claim 9,

step (1) including the step of incrementing a program counter with the first instruction; and

step (3) including the step of generating a second probe signal indicating a content of the program counter.

11. The method of claim 9,

before step (5),

searching a second memory device for an entry associated with the first address,

when the entry in the second memory device does not exist, generating at least one probe signal indicating a miss entry in the second memory device, and

generating a temporal identifier signal that is associated with the probe signal.

12. The method of claim 9,

before step (2),

searching an address storage device for an entry associated with the first address,

when the entry in the address storage device does not exist, generating at least one probe signal indicating a miss entry in the address storage memory device, and

generating a temporal identifier signal that is associated with the probe signal.

13. A method for monitoring an execution behavior of a program, comprising:

generating probe signals representative of memory access misses occurring in a processor;

receiving the probe signals and associating a temporal identifier signal with the probe signals; and

storing the temporal identifier signal and the probe signals; and

generating a second high-speed memory miss signal, a second high-speed memory miss count signal and a time stamp signal, the second high-speed memory miss signal indicating a miss in a second high-speed memory, the second high-speed memory miss count signal representing a number of misses in the second high-speed memory, and the time stamp signal indicating when the second high-speed memory miss signal is active.

14. The method of claim 13, further comprising:

generating probe signals in response to a memory access miss signal when executing a specified instruction.

15. The method of claim 13, further comprising:

generating probe signals recording a TLB miss when executing a specified instruction.

16. The method of claim 13, further comprising

generating a program counter signal, a TLB identification signal indicating a miss in the TLB, a TLB miss count signal representing an accumulative count of TLB misses, and a time stamp signal when the TLB miss signal is activated.

17. The method of claim 16, further comprising

incrementing a TLB miss counter when the TLB miss signal is activated.

18. The method of claim 13, wherein

the program executes on a processor;

the processor includes a first high-speed memory and a program counter, the first high-speed memory generating a first high-speed memory miss signal; and

a probe logic unit generates a program counter signal, a first high-speed memory miss signal indicating a miss in the first high-speed memory, a first high-speed memory miss count signal representing a number of misses in the first high-speed memory, and a time stamp signal when the first high-speed memory miss signal is activated.

19. The method of claim 18, wherein

the probe logic unit includes a first high-speed memory miss counter coupled to the first high-speed memory miss signal, the first high-speed memory miss counter is incremented when the first high-speed memory miss signal is activated.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None